

13. 7. 2004

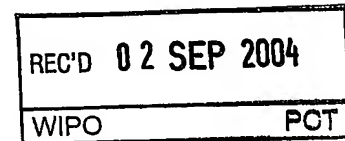
日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 3 年 7 月 1 1 日

出 願 番 号  
Application Number: 特 願 2 0 0 3 - 2 7 3 3 7 8  
[ST. 10/C]: [ J P 2 0 0 3 - 2 7 3 3 7 8 ]



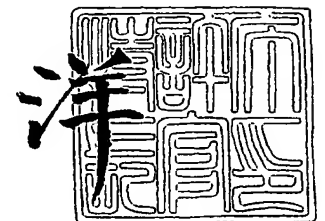
出 願 人  
Applicant(s): 松下電器産業株式会社

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1 (a) OR (b)

2 0 0 4 年 8 月 1 9 日

特許庁長官  
Commissioner,  
Japan Patent Office

小 川



【書類名】 特許願  
【整理番号】 2900655348  
【提出日】 平成15年 7月11日  
【あて先】 特許庁長官殿  
【国際特許分類】 H03L 1/00  
【発明者】  
    【住所又は居所】 神奈川県横浜市港北区綱島東四丁目3番1号 パナソニックモバイルコミュニケーションズ株式会社内  
    【氏名】 李 継峰  
【特許出願人】  
    【識別番号】 000005821  
    【氏名又は名称】 松下電器産業株式会社  
【代理人】  
    【識別番号】 100105050  
    【弁理士】  
    【氏名又は名称】 鷺田 公一  
【手数料の表示】  
    【予納台帳番号】 041243  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9700376

**【書類名】 特許請求の範囲****【請求項 1】**

所定サイズのウィンドウに対して複数の処理系統で並列に復号演算を行う復号装置であって、

前記ウィンドウ内において前記複数の処理系統数だけ前の時点の前方確率から現時点の前方確率を順次演算する前方確率演算手段と、

前記ウィンドウ内において前記複数の処理系統数だけ後の時点の後方確率から現時点の後方確率を順次演算する後方確率演算手段と、

前記前方確率および前記後方確率を用いて尤度情報を演算する尤度演算手段と、  
を有することを特徴とする復号装置。

**【請求項 2】**

前記後方確率演算手段は、

前記ウィンドウより後の時点のデータをトレーニングデータとして現時点の後方確率を演算することを特徴とする請求項 1 記載の復号装置。

**【請求項 3】**

前記後方確率演算手段は、

前記複数の処理系統で共通のトレーニングデータを用いることを特徴とする請求項 2 記載の復号装置。

**【請求項 4】**

請求項 1 から請求項 3 のいずれかに記載の復号装置を有することを特徴とする基地局装置。

**【請求項 5】**

請求項 1 から請求項 3 のいずれかに記載の復号装置を有することを特徴とする移動局装置。

**【請求項 6】**

所定サイズのウィンドウに対して複数の処理系統で並列に復号演算を行う復号方法であって、

前記ウィンドウ内において前記複数の処理系統数だけ後の時点の後方確率から現時点の後方確率を順次演算するステップと、

前記ウィンドウ内において前記複数の処理系統数だけ前の時点の前方確率から現時点の前方確率を順次演算するステップと、

前記前方確率が演算されるごとに前記前方確率および前記後方確率を用いて尤度情報を演算するステップと、

を有することを特徴とする復号方法。

【書類名】明細書

【発明の名称】復号装置および復号方法

【技術分野】

【0001】

本発明は、復号装置および復号方法に関し、特に、Max-Log-MAP アルゴリズムを用いたターボ復号を行う復号装置および復号方法に関する。

【背景技術】

【0002】

近年、第四世代移動体通信に採用される方式の最も有力な候補として、VSF-OFCDM (Variable Spreading Factor-Orthogonal Frequency and Code Division Multiplexing: 可変拡散率直交周波数・符号分割多重) が注目されている。VSF-OFCDM が採用された場合には、およそ 50-100 MHz の帯域幅を用いて、100 Mbps 以上の最大伝送速度を実現することが可能となる。このような超高速な通信方式には、誤り訂正方式としてターボ符号の適用が有効である。

【0003】

ターボ符号方式は、送信データに対して、畳み込み符号化とインタリーブを併用し、復号時に繰り返し復号することの特徴としている。繰り返し復号処理をすることにより、ランダム誤りはもちろんのこと、バースト誤りに対しても優れた誤り訂正能力を示すことが知られている。

【0004】

以下、ターボ復号の手順について簡単に説明する。

【0005】

ターボ復号の処理手順は、大きく前方確率算出、後方確率算出、および尤度情報算出に分けられる。

【0006】

前方確率  $\alpha$  の算出は、下記の式 (1) により状態ごとに行われる。

【0007】

【数 1】

$$\log \alpha_k(m) = \log \sum_{m' \in (m' \rightarrow m)} e^{\log \alpha_{k-1}(m') + \log \gamma_k(b)} \quad \dots \text{式 (1)}$$

上式 (1) において、 $\log \alpha$  は対数領域での前方確率、 $k$  は時点、 $m$  および  $m'$  は状態遷移トレリス上における状態をそれぞれ示している。すなわち、式 (1) の左辺は、時点  $k$  の状態  $m$  における前方確率を自然対数で示している。また、上式 (1) において、 $\log \gamma$  は対数領域での移行確率、 $m' \in (m' \rightarrow m)$  は状態  $m$  へ遷移可能なすべての状態  $m'$ 、 $b$  は送信信号の組み合わせ、すなわちシステムチックビットとパリティビットの取り得る組み合わせを示している。

【0008】

式 (1) から明らかなように、前方確率  $\alpha_k$  は、前時点 ( $k-1$ ) における前方確率  $\alpha_{k-1}$  から算出される。

【0009】

次に、後方確率  $\beta$  の算出は、下記の式 (2) により状態ごとに行われる。

【0010】

【数 2】

$$\log \beta_k(m) = \log \sum_{m' \in (m' \rightarrow m)} e^{\log \beta_{k+1}(m') + \log \gamma_{k+1}(b)} \quad \dots \text{式 (2)}$$

計算法は前方確率とはほぼ同様であるが、大きく異なる点は、後方確率  $\beta_k$  は、後時点 ( $k+1$ ) における後方確率  $\beta_{k+1}$  から算出される点である。つまり、前方確率は、時間軸上の順方向に計算していくのに対して、後方確率は、時間軸上の逆方向に計算していくことになる。

【0011】

次に、尤度情報  $L(u_k)$  の算出は、下記の式 (3) によって行われる。

【0012】

【数3】

$$L(u_k) = \log \frac{\sum_{u_k=0} e^{\log \alpha_{k-1}(m') + \log \beta_k(m) + \log \gamma_k(b)}}{\sum_{u_k=1} e^{\log \alpha_{k-1}(m') + \log \beta_k(m) + \log \gamma_k(b)}} \quad \dots \text{式 (3)}$$

式 (3) において、分子は送信信号中のシステムチックビット  $u_k = 0$  となるすべての状態遷移の組み合わせにおける演算を表し、分母は  $u_k = 1$  となるすべての状態遷移の組み合わせにおける演算を表す。

【0013】

上式 (1) ~ (3) の計算は、非常に煩雑であるため、ターボ復号の1つのアルゴリズムである  $\text{Max-Log-MAP}$  アルゴリズムにおいては、式 (4) に示す近似式が用いられる。

【0014】

【数4】

$$\log(e^A + e^B) = \max(A, B) + \log(1 + e^{-|A-B|}) \quad \dots \text{式 (4)}$$

式 (4) を用いて式 (1) および式 (2) を変形すると、それぞれ以下の式 (5) および式 (6) のようになる。

【0015】

【数5】

$$\alpha_k(m) = \max_{m'} (\alpha_{k-1}(m') + \gamma_{k-1}(m', m)) \quad \dots \text{式 (5)}$$

$$\beta_k(m) = \max_{m'} (\beta_{k+1}(m') + \gamma_{k+1}(m', m)) \quad \dots \text{式 (6)}$$

さらに、これらの式 (5), (6) を用いて式 (3) を変形すると、以下の式 (7) のようになる。

【0016】

【数6】

$$L(u_k) = \max_{(m', m)_{u_k=0}} (\alpha_{k-1}(m') + \beta_k(m) + \gamma_k(m', m)) - \max_{(m', m)_{u_k=1}} (\alpha_{k-1}(m') + \beta_k(m) + \gamma_k(m', m)) \quad \dots \text{式 (7)}$$

$\text{Max-Log-MAP}$  アルゴリズムを用いたターボ復号においては、式 (7) を用いて算出された尤度情報  $L(u_k)$  が閾値 0 と比較され、尤度情報  $L(u_k)$  が 0 以上であれば、時点  $k$  で送信されたシステムチックビット  $u_k = 1$  と硬判定され、尤度情報  $L(u_k)$  が 0 未満であれば、時点  $k$  で送信されたシステムチックビット  $u_k = 0$  と硬判定される。

【0017】

ここで、式 (7) に示したように、時点  $k$  における尤度情報算出のためには、時点  $(k-1)$  における前方確率  $\alpha_{k-1}$ 、ならびに時点  $k$  における後方確率  $\beta_k$  および移行確率  $\gamma_k$  が必要となる。このとき、時点  $1 \sim k$  における各時点での前方確率および後方確率をそれぞれ算出し、その後、尤度情報を算出する場合は、全状態の全時点での確率値を記憶する必要があるため、メモリ量は膨大なものとなる。

【0018】

このメモリ量を削減するためには、例えば以下のような算出手順を取ることが考えられる。すなわち、まず後方確率を時点  $k \sim 1$  について算出し、メモリへ蓄積する。次に時点ごとに前方確率を算出し、この前方確率および先に算出した後方確率より、尤度情報を逐次的に算出する。この方法によれば、算出された前方確率が即座に尤度情報の算出に用いられるため、前方確率の蓄積は行われず、前方確率の蓄積の分だけメモリ量を削減することができる。

【0019】

また、後方確率を蓄積するためのメモリ量の削減方法として、例えば非特許文献1に記載されたスライディングウィンドウ法がある。スライディングウィンドウ法は、データの全系列を所定のウィンドウ単位に分割し、各ウィンドウにトレーニング区間を設けることにより、系列の最後方から計算しなければならない後方確率を、系列の中途から計算する方法である。このスライディングウィンドウ法によれば、ウィンドウ単位で後方確率を蓄積すれば良く、時点 $k-1$ のすべての後方確率を蓄積する場合に比べてメモリ量を大幅に削減することができる。

#### 【0020】

さらに、スライディングウィンドウ法において、確率値および尤度情報の演算をパラレルに行うことにより、演算の高速化を図ることができる。すなわち、例えば図5(a)に示すように、全系列 $nW$ のデータが $n$ 個のウィンドウに分割される場合、このウィンドウを並列に処理することにより、演算の高速化を図ることができる。例えば図5(b)に示すように、#1および#2の2つの処理系統によって並列に演算を行うことにより、演算時間を半分にすることができる。

【非特許文献1】Andrew J. Viterbi, "An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes", IEEE J. Sel. Areas Commun., vol.16, no.2, pp.260-264, Feb.1998

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0021】

しかしながら、上述のようにパラレルにスライディングウィンドウ法を行う場合でも、ウィンドウサイズに対応する処理遅延が生じるという問題がある。特に、Max-Log-MAPアルゴリズムを用いて尤度情報 $L(u_k)$ を算出する場合、時点 $k$ が小さい方から順に算出する必要があるが、尤度情報 $L(u_k)$ の算出に必要な確率値は、各ウィンドウ内ではシリアルに算出されるため、ウィンドウを並列に処理しても、結局尤度情報 $L(u_k)$ の算出には遅延が生じることになる。

#### 【0022】

この問題について、図6を参照して具体的に説明する。

#### 【0023】

図6は、スライディングウィンドウ法のウィンドウサイズが64である場合に、2つの処理系統#1、#2によって尤度情報 $L(u_k)$ を算出するタイミングの例を示す図である。図6(a)は、後方確率 $\beta_k$ を算出するタイミングを示し、図6(b)は、前方確率 $\alpha_k$ を算出するタイミングを示し、図6(c)は、尤度情報 $L(u_k)$ を算出するタイミングを示している。

#### 【0024】

図6(a)において、後方確率 $\beta_{63} \sim \beta_0$ および後方確率 $\beta_{127} \sim \beta_{64}$ を算出するために、それぞれトレーニング区間が設けられている。トレーニング区間の先頭(ここでは、 $\beta_{95}$ および $\beta_{159}$ )は、常に0と仮定され、トレーニング区間内で上記の式(6)の演算が行われることにより、ウィンドウ内の後方確率 $\beta_{63} \sim \beta_0$ および後方確率 $\beta_{127} \sim \beta_{64}$ を正しく算出することができる。したがって、トレーニング区間としては、少なくとも20程度のサイズが必要となる。

#### 【0025】

図6(a)においては、時刻 $T1$ で後方確率 $\beta_0$ 、 $\beta_{64}$ の算出が完了する。各処理系統#1、#2は、引き続き後方確率 $\beta_{191} \sim \beta_{128}$ および後方確率 $\beta_{255} \sim \beta_{192}$ を算出する。

#### 【0026】

そして、図6(b)に示すように、時刻 $T1$ から前方確率 $\alpha_0 \sim \alpha_{63}$ および $\alpha_{64} \sim \alpha_{127}$ の算出が開始される。このとき、前方確率 $\alpha_{64} \sim \alpha_{127}$ の算出に関しては、 $\alpha_{32}$ を0と仮定し、 $\alpha_{32} \sim \alpha_{63}$ がトレーニング区間となっている。

#### 【0027】

一方、図6(c)に示すように、時刻 $T1$ から尤度情報 $L(u_0) \sim L(u_{63})$ の算出

が開始される。時刻  $T_1$  においては、後方確率  $\beta_0 \sim \beta_{63}$  は既に算出されており、前方確率  $\alpha_0$  も算出されるため、処理系統 # 1 は、尤度情報  $L(u_0)$  を算出することができる。

【0028】

しかし、時刻  $T_1$  において、後方確率  $\beta_{64} \sim \beta_{127}$  が既に算出されており、前方確率  $\alpha_{64}$  も算出されているにも拘わらず、処理系統 # 2 は、尤度情報  $L(u_{64})$  を算出することができずに待機する。これは、上述したように尤度情報  $L(u_k)$  は、時点  $k$  が小さい方から順に算出される必要があるが、時刻  $T_1$  においては、尤度情報  $L(u_0)$  が算出されるのみであり、尤度情報  $L(u_{63})$  は後になって処理系統 # 1 によって算出されるためである。

【0029】

したがって、図 6 (c) に示すように、時刻  $T_1$  から処理系統 # 1 によって尤度情報  $L(u_0) \sim L(u_{63})$  が算出され、時刻  $T_2$  から処理系統 # 2 によって尤度情報  $L(u_{64}) \sim L(u_{127})$  が順次算出される。

【0030】

つまり、後方確率および前方確率の算出は完了しているにも拘わらず、尤度情報の算出に際して並列処理を行うことができず、ウィンドウサイズに対応する処理遅延が生じてしまう。この処理遅延を小さくするためには、ウィンドウサイズを小さくすれば良いが、ウィンドウサイズを小さくした場合には、ウィンドウ数が増加することになり、これに伴ってトレーニング区間に対する処理量が増大する。トレーニング区間は、実際の復号には寄与しない区間であるため、トレーニング区間に対する処理量が増大すれば、全体の処理量は増大し、結果として回路規模が増大することがある。

【0031】

本発明は、かかる点に鑑みてなされたものであり、処理量および回路規模の増大を抑制しつつ、高速に尤度情報を算出することができる復号装置および復号方法を提供することを目的とする。

【課題を解決するための手段】

【0032】

本発明の復号装置は、所定サイズのウィンドウに対して複数の処理系統で並列に復号演算を行う復号装置であって、前記ウィンドウ内において前記複数の処理系統数だけ前の時点の前方確率から現時点の前方確率を順次演算する前方確率演算手段と、前記ウィンドウ内において前記複数の処理系統数だけ後の時点の後方確率から現時点の後方確率を順次演算する後方確率演算手段と、前記前方確率および前記後方確率を用いて尤度情報を演算する尤度演算手段と、を有する構成を採る。

【0033】

この構成によれば、処理系統数だけ前後の時点の前方確率および後方確率から現時点の前方確率および後方確率を演算するため、各ウィンドウ内においても複数の処理系統が並列に確率値の演算を行うことができ、確率値が演算されるごとに後段の尤度演算を行って、処理量および回路規模の増大を抑制しつつ、高速に尤度情報を算出することができる。

【0034】

本発明の復号装置は、前記後方確率演算手段は、前記ウィンドウより後の時点のデータをトレーニングデータとして現時点の後方確率を演算する構成を採る。

【0035】

この構成によれば、処理対象のウィンドウより後の時点のデータをトレーニングデータとして現時点の後方確率を演算するため、任意のウィンドウに関して後方確率の演算処理を行うことができる。

【0036】

本発明の復号装置は、前記後方確率演算手段は、前記複数の処理系統で共通のトレーニングデータを用いる構成を採る。

【0037】

この構成によれば、複数の処理系統で共通のトレーニングデータを用いるため、トレー

ニングデータに必要な演算量を削減することができ、装置全体の処理量をさらに削減することができる。

【0038】

本発明の基地局装置は、上記のいずれかに記載の復号装置を有する構成を採る。

【0039】

この構成によれば、上記のいずれかに記載の復号装置と同様の作用効果を基地局装置において実現することができる。

【0040】

本発明の移動局装置は、上記のいずれかに記載の復号装置を有する構成を採る。

【0041】

この構成によれば、上記のいずれかに記載の復号装置と同様の作用効果を移動局装置において実現することができる。

【0042】

本発明の復号方法は、所定サイズのウィンドウに対して複数の処理系統で並列に復号演算を行う復号方法であって、前記ウィンドウ内において前記複数の処理系統数だけ後の時点の後方確率から現時点の後方確率を順次演算するステップと、前記ウィンドウ内において前記複数の処理系統数だけ前の時点の前方確率から現時点の前方確率を順次演算するステップと、前記前方確率が演算されるごとに前記前方確率および前記後方確率を用いて尤度情報を演算するステップと、を有するようにした。

【0043】

この方法によれば、処理系統数だけ前後の時点の前方確率および後方確率から現時点の前方確率および後方確率を演算するため、各ウィンドウ内においても複数の処理系統が並列に確率値の演算を行うことができ、確率値が演算されるごとに後段の尤度演算を行って、処理量および回路規模の増大を抑制しつつ、高速に尤度情報を算出することができる。

【発明の効果】

【0044】

本発明によれば、処理量および回路規模の増大を抑制しつつ、高速に尤度情報を算出することができる。

【発明を実施するための最良の形態】

【0045】

本発明の骨子は、ウィンドウ単位で並列処理を行うとともに、ウィンドウ内における演算処理も並列に行うことである。

【0046】

以下、本発明の一実施の形態について、図面を参照して詳細に説明する。

【0047】

図1は、本発明の一実施の形態に係る復号装置の全体構成を示すブロック図である。同図に示す復号装置は、インタリーバ100、復号器110、インタリーバ120、復号器130、デインタリーバ140、デインタリーバ150、硬判定部160、および誤り検出部170を有している。なお、図1に示す復号装置は、ターボ符号化によって得られる情報ビットそのものであるシステムチックビット $x_a$ 、情報ビットを畳み込み符号化して得られるパリティビット $x_b$ 、および情報ビットをインタリーブ後に畳み込み符号化して得られるパリティビット $x_c$ に、それぞれ通信路における雑音が付加されたシステムチックビット $y_a$ 、パリティビット $y_b$ 、およびパリティビット $y_c$ を復号するものとする。

【0048】

インタリーバ100は、システムチックビット $y_a$ をインタリーブする。インタリーバ100によるインタリーブは、符号化側におけるインタリーブと同様に行われる。

【0049】

復号器110は、軟入力軟出力の復号器であり、前回の復号結果から得られる事前情報尤度 $L_e$ とシステムチックビット $y_a$ とパリティビット $y_b$ とを用いて復号を行い、復号結果をインタリーバ120へ出力する。復号器110による復号については、後に詳述する



## 【0050】

インタリーブ120は、復号器110の復号結果をインタリーブする。インタリーブ120によるインタリーブは、符号化側におけるインタリーブと同様に行われる。

## 【0051】

復号器130は、軟入力軟出力の復号器であり、インタリーブ120から出力される事前情報尤度とインタリーブ100から出力されるシステムチェックビット $y_a$ とパリティビット $y_c$ を用いて復号を行い、復号結果をデインタリーブ140およびデインタリーブ150へ出力する。

## 【0052】

デインタリーブ140およびデインタリーブ150は、復号器130の復号結果をデインタリーブする。デインタリーブ140およびデインタリーブ150によるデインタリーブは、インタリーブ100またはインタリーブ120によるインタリーブを元に戻すようにして行われる。

## 【0053】

硬判定部160は、デインタリーブ150から出力される尤度情報を硬判定し、0または1の硬判定値を出力する。具体的には、硬判定部160は、デインタリーブ150から出力される尤度情報を閾値である0と比較し、尤度情報が0以上であれば硬判定値として1を出力し、尤度情報が0未満であれば硬判定値として0を出力する。

## 【0054】

誤り検出部170は、硬判定結果に付加されているCRC (Cyclic Redundancy Check) などの誤り検出符号を用いて誤り検出を行い、復号データを出力する。

## 【0055】

次いで、図2および図3を用いて、上記のように構成された復号装置における復号器の動作について具体的に説明する。

## 【0056】

図2は、復号器110の内部構成を示すブロック図である。同図に示すように、復号器110は、移行確率演算部111、後方確率演算部112、前方確率演算部113、記憶部114、および尤度演算部115を有している。なお、復号器130も復号器110と同様の内部構成を有しているものとする。また、以下の復号動作は、所定サイズのウィンドウ単位で行われるものとする。

## 【0057】

まず、前回の復号結果から得られる事前情報尤度 $L_e$ 、システムチェックビット $y_a$ 、およびパリティビット $y_b$ が移行確率演算部111へ入力され、移行確率が演算される。以下では、時点 $k$ において状態遷移トレリス上の状態 $m'$ から状態 $m$ へ移行する移行確率を $\gamma_k(m', m)$ と表記する。

## 【0058】

算出された移行確率 $\gamma_k(m', m)$ は、後方確率演算部112および前方確率演算部113へ出力され、それぞれ後方確率および前方確率が算出される。以下では、時点 $k$ の状態 $m$ における後方確率を $\beta_k(m)$ と表記し、時点 $k$ の状態 $m$ における前方確率を $\alpha_k(m)$ と表記する。

## 【0059】

ここで、後方確率演算部112および前方確率演算部113による確率値の算出について説明する。

## 【0060】

上述したように、前方確率 $\alpha_k(m)$ および後方確率 $\beta_k(m)$ は、式(5)、(6)によって算出される。式(5)、(6)は、それぞれ以下の式(8)、(9)のように変形することができる。

## 【0061】

## 【数7】

$$\alpha_{k-1}(m) = \max_{m'} (\alpha_{k-2}(m') + \gamma_{k-2}(m', m)) \quad \dots \text{式 (8)}$$

$$\beta_{k+1}(m) = \max_{m'} (\beta_{k+2}(m') + \gamma_{k+2}(m', m)) \quad \dots \text{式 (9)}$$

これらの式(8), (9)を再度式(5), (6)へ代入すると、以下の式(10), (11)が得られる。

## 【0062】

## 【数8】

$$\begin{aligned} \alpha_k(m) &= \max_{m'} \left( \max_{m''} (\alpha_{k-2}(m'') + \gamma_{k-2}(m'', m')) + \gamma_{k-1}(m', m) \right) \\ &= \max_{m'} \left( \alpha_{k-2}(m'') + \max_{m'} (\gamma_{k-2}(m'', m') + \gamma_{k-1}(m', m)) \right) \\ &= \max_{m''} (\alpha_{k-2}(m'') + \mu_{k-2}(m'', m)) \quad \dots \text{式 (10)} \end{aligned}$$

$$\text{ただし、} \mu_{k-2}(m'', m) = \max_{m'} (\gamma_{k-2}(m'', m') + \gamma_{k-1}(m', m))$$

$$\begin{aligned} \beta_k(m) &= \max_{m'} \left( \max_{m''} (\beta_{k+2}(m'') + \gamma_{k+2}(m'', m')) + \gamma_{k+1}(m', m) \right) \\ &= \max_{m''} \left( \beta_{k+2}(m'') + \max_{m'} (\gamma_{k+2}(m'', m') + \gamma_{k+1}(m', m)) \right) \\ &= \max_{m''} (\beta_{k+2}(m'') + \eta_{k+2}(m'', m)) \quad \dots \text{式 (11)} \end{aligned}$$

$$\text{ただし、} \eta_{k+2}(m'', m) = \max_{m'} (\gamma_{k+2}(m'', m') + \gamma_{k+1}(m', m))$$

式(10)は、時点kの前方確率 $\alpha_k$ を時点(k-2)の前方確率 $\alpha_{k-2}$ から算出することを示し、式(11)は、時点kの後方確率 $\beta_k$ を時点(k+2)の後方確率 $\beta_{k+2}$ から算出することを示している。

## 【0063】

後方確率演算部112および前方確率演算部113は、それぞれ式(10), (11)を用いて、2つの処理系統で並列に後方確率および前方確率を演算する。つまり、例えば後方確率演算部112における後方確率の演算においては、一方の処理系統が後方確率 $\beta_{k+2}$ から後方確率 $\beta_k$ を算出する間に、他方の処理系統が並行して後方確率 $\beta_{k+1}$ から後方確率 $\beta_{k-1}$ を算出する。具体的にk=1の場合を考えると、2つの処理系統によって、後方確率 $\beta_1$ および後方確率 $\beta_0$ が並行して算出されることになる。

## 【0064】

算出された後方確率は、ウィンドウ単位で記憶部114へ記憶される。また、後方確率と同様に、前方確率演算部113においても、2つの処理系統によって、前方確率 $\alpha_k$ と前方確率 $\alpha_{k+1}$ が並行して算出される。前方確率が算出されると、尤度演算部115によって、前方確率および記憶部114に記憶されている後方確率が用いられ、上述した式(7)によって尤度情報が算出される。

## 【0065】

このとき、従来とは異なり、前方確率演算部113における2つの処理系統は、前方確率 $\alpha_k$ および前方確率 $\alpha_{k+1}$ を並行して算出するため、時点kが小さい方から順に尤度情報L( $u_k$ )を算出する場合に、双方の処理系統によって算出される前方確率を同時に用いて尤度情報を算出することができる。

## 【0066】

以下、図3を参照して、尤度情報を算出するタイミングについて、具体的に説明する。

## 【0067】

図3(a)は、後方確率演算部112における2つの処理系統#1, #2による後方確率 $\beta_k$ の算出タイミングを示している。同図に示すように、処理系統#1は、 $\beta_{190} \sim \beta_{128}$ をトレーニング区間として、後方確率 $\beta_{126} \sim \beta_0$ のうち時点kが偶数であるものを順次算出していく。同様に、処理系統#2は、 $\beta_{191} \sim \beta_{129}$ をトレーニング区間として、後方確率 $\beta_{127} \sim \beta_1$ のうち時点kが奇数であるものを順次算出していく。そして、時刻T1に

において、後方確率  $\beta_0$  および後方確率  $\beta_1$  が算出される。

【0068】

図3 (b) は、前方確率演算部 113 における2つの処理系統#1, #2による前方確率  $\alpha_k$  の算出タイミングを示している。同図に示すように、処理系統#1は、時刻T1から前方確率  $\alpha_0 \sim \alpha_{126}$  のうち時点kが偶数であるものを順次算出していく。同様に、処理系統#2は、時刻T1から前方確率  $\alpha_1 \sim \alpha_{127}$  のうち時点kが奇数であるものを順次算出していく。

【0069】

図3 (c) は、尤度演算部 115 における2つの処理系統#1, #2による尤度情報L ( $u_k$ ) の算出タイミングを示している。同図に示すように、処理系統#1は、時刻T1において尤度情報L ( $u_0$ ) を算出する。そして、処理系統#2は、時刻T1において前方確率  $\alpha_0$  および既に記憶部 114 に記憶されている後方確率  $\beta_1$  を用いて尤度情報L ( $u_1$ ) を算出する。

【0070】

このように従来とは異なり、処理系統#2は尤度情報L ( $u_k$ ) の算出を待機することが無く、並行処理を行うことができる。このため、図3 (c) と図6 (c) とを比較すれば明らかなように、尤度情報L ( $u_k$ ) の演算を大幅に高速化することができる。

【0071】

このように、本実施の形態によれば、処理系統数(ここでは2)だけ前または後の時点の確率値から現時点での確率値を算出する演算を、各処理系統で並列に行うため、確率値を用いた尤度情報の算出における処理遅延が無く、処理量および回路規模の増大を抑制しつつ、高速に尤度情報を算出することができる。

【0072】

なお、本実施の形態においては、トレーニング区間のサイズを32とし、ウィンドウサイズを64として説明したが、本発明はこれらのサイズに限定されない。トレーニング区間のサイズとしては、約20程度以上であれば良い。

【0073】

また、本実施の形態においては、2つの処理系統で並列に演算する場合について説明したが、処理系統数は2以上であればいくつでも良い。この場合には、式(10), (11)と同様の考え方で、前方確率および後方確率を処理系統数だけ前および後の時点の前方確率および後方確率で表す式を用いれば良い。すなわち、処理系統数をm (mは2以上の整数) とすれば、前方確率  $\alpha_k$  および後方確率  $\beta_k$  をそれぞれ前方確率  $\alpha_{k-m}$  および後方確率  $\beta_{k+m}$  で表す式を用いて確率値を演算すれば良い。

【0074】

また、例えば図4に示すように、トレーニング区間についてはいずれか1つの処理系統のみにおいて連続した演算を行い、このトレーニング結果をすべての処理系統で共通して用いることにより、装置全体の演算量をさらに削減することができる。

【図面の簡単な説明】

【0075】

【図1】 本発明の一実施の形態に係る復号装置の全体構成を示すブロック図

【図2】 一実施の形態に係る復号器の内部構成を示すブロック図

【図3】 一実施の形態に係る復号動作のタイミングの例を示す図

【図4】 一実施の形態に係る復号動作のタイミングの他の例を示す図

【図5】 ウィンドウを用いた処理を説明するための図

【図6】 スライディングウィンドウ法による復号動作のタイミングの例を示す図

【符号の説明】

【0076】

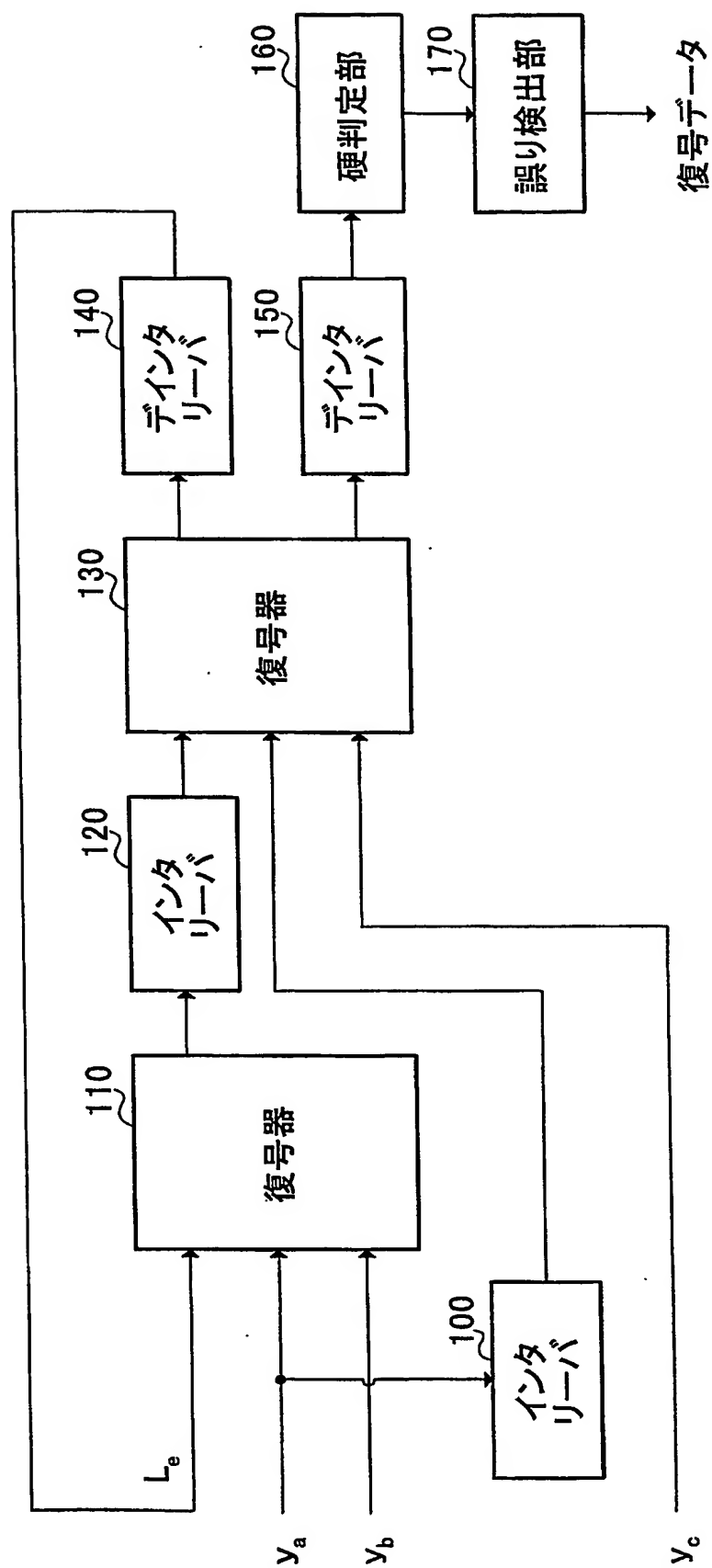
100、120 インタリバー

110 復号器

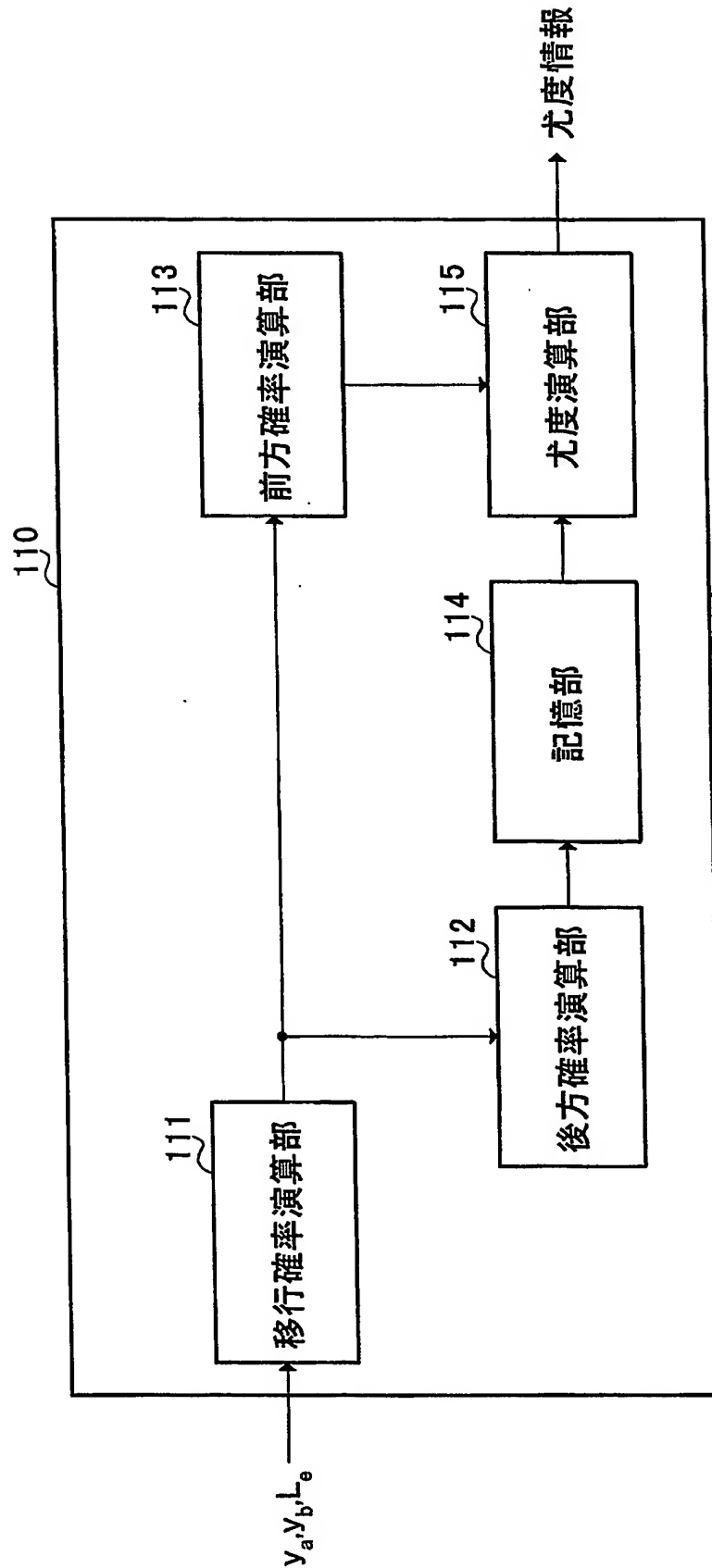
111 移行確率演算部

- 1 1 2 後方確率演算部
- 1 1 3 前方確率演算部
- 1 1 4 記憶部
- 1 1 5 尤度演算部
- 1 3 0 復号器
- 1 4 0、1 5 0 デインタリーバ
- 1 6 0 硬判定部
- 1 7 0 誤り検出部

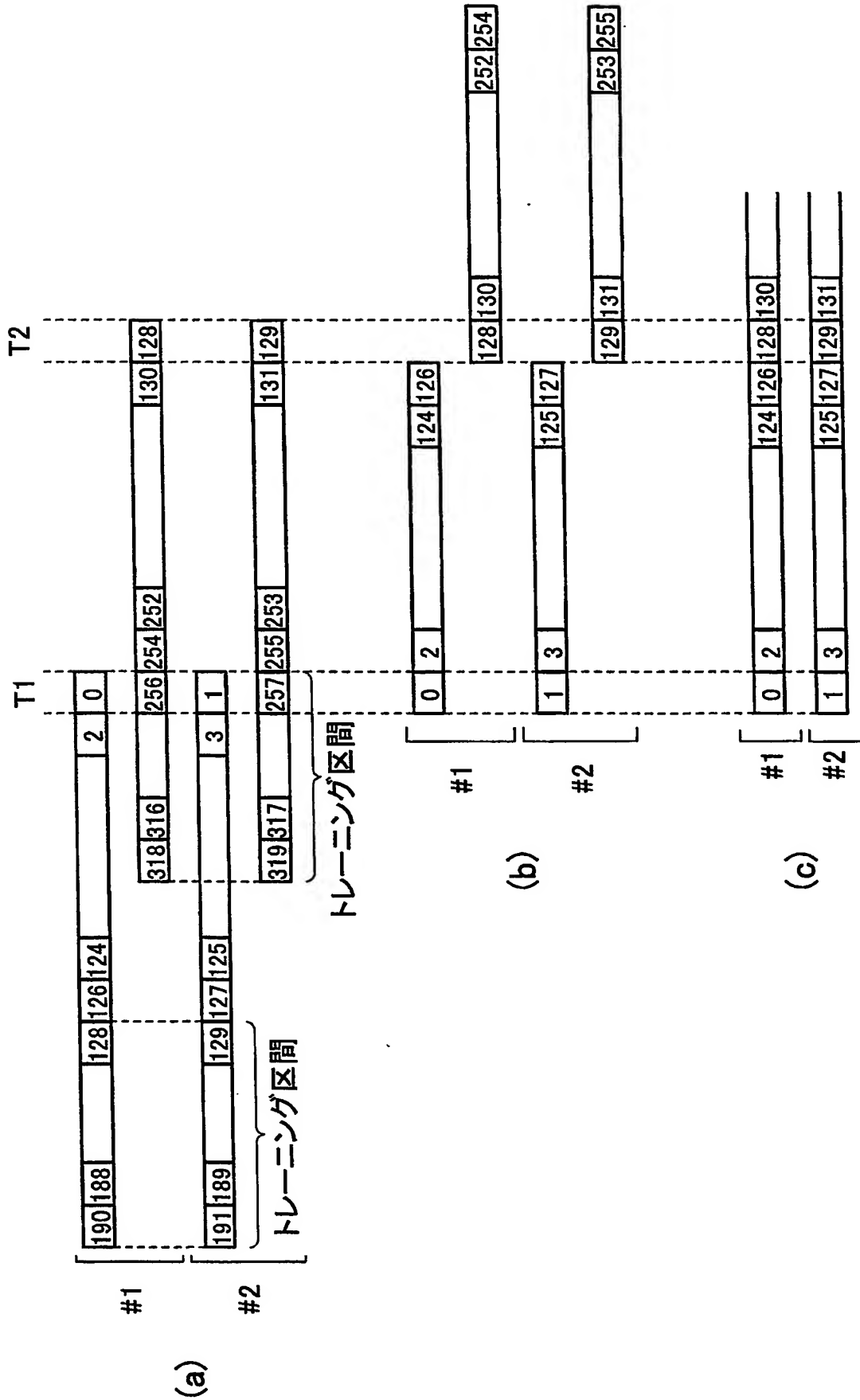
【書類名】 図面  
【図 1】



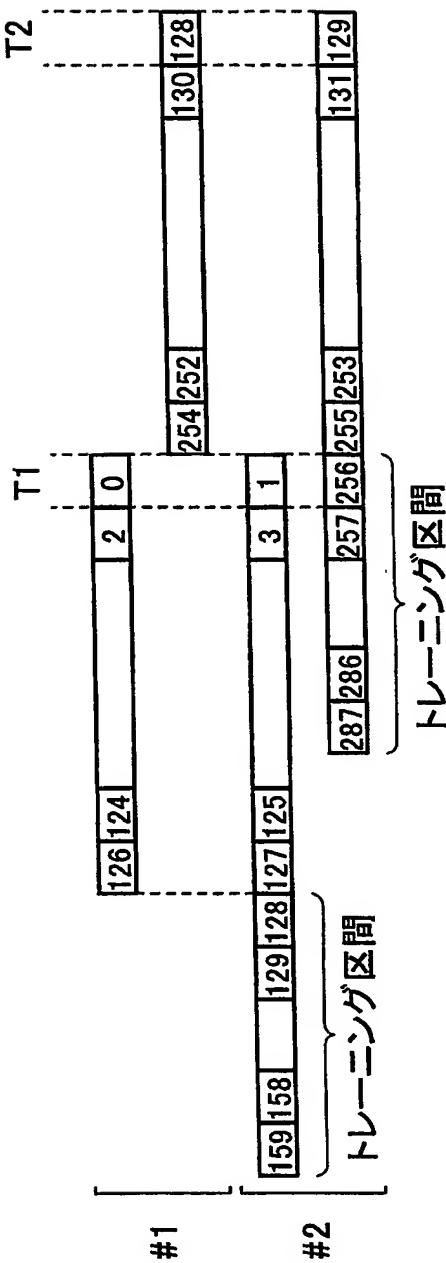
【図 2】



【図 3】

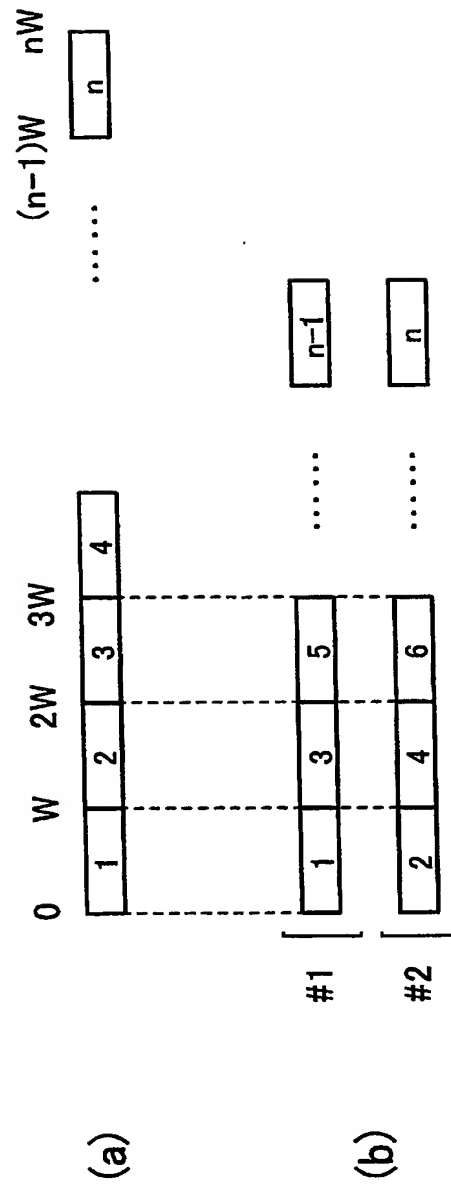


【図 4】

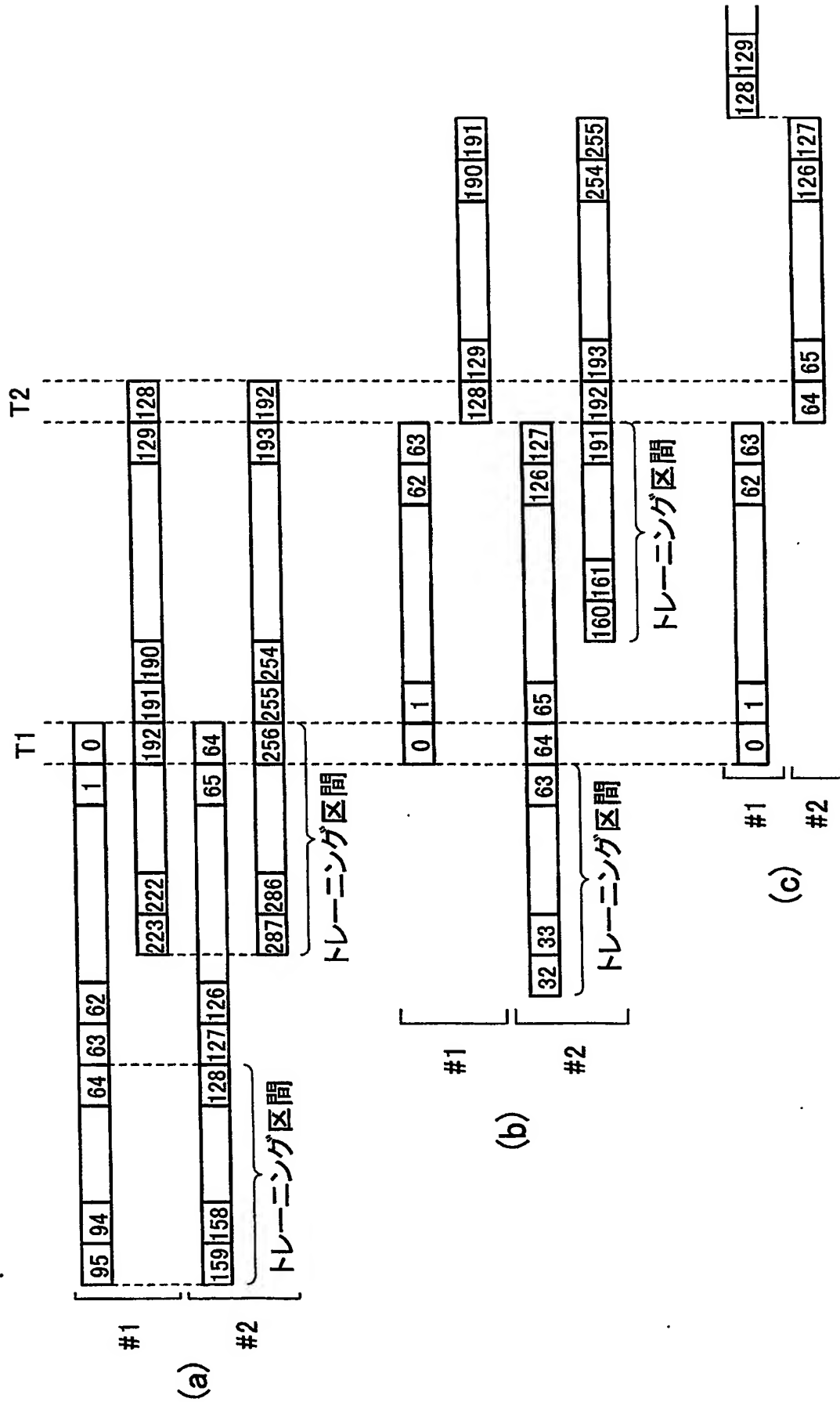




【図 5】



【図 6】



## 【書類名】要約書

## 【要約】

【課題】 処理量および回路規模の増大を抑制しつつ、高速に尤度情報を算出すること。

【解決手段】 後方確率演算部 112 における後方確率の演算においては、一方の処理系統が後方確率  $\beta_{k+2}$  から後方確率  $\beta_k$  を算出する間に、他方の処理系統が並行して後方確率  $\beta_{k+1}$  から後方確率  $\beta_{k-1}$  を算出する。具体的に  $k=1$  の場合を考えると、2つの処理系統によって、後方確率  $\beta_1$  および後方確率  $\beta_0$  が並行して算出されることになる。算出された後方確率は、ウィンドウ単位で記憶部 114 へ記憶される。また、後方確率と同様に、前方確率演算部 113 においても、2つの処理系統によって、前方確率  $\alpha_k$  と前方確率  $\alpha_{k+1}$  が並行して算出される。前方確率が算出されると、尤度演算部 115 によって、前方確率および記憶部 114 に記憶されている後方確率が用いられ、尤度情報が算出される。

【選択図】 図 2

特願 2 0 0 3 - 2 7 3 3 7 8

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 5 8 2 1 ]

1. 変更年月日	1 9 9 0 年 8 月 2 8 日
[変更理由]	新規登録
住 所	大阪府門真市大字門真 1 0 0 6 番地
氏 名	松下電器産業株式会社

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**